Data Structures and Algorithms

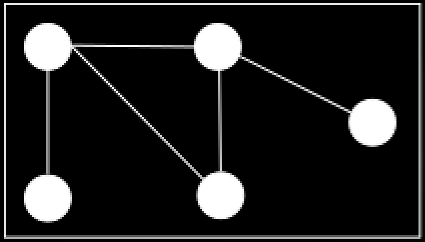
Lecture 10

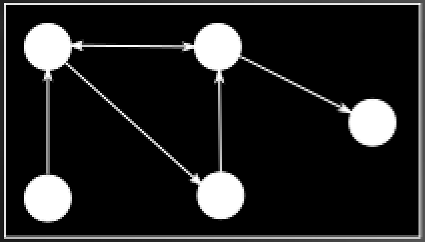
Computer programming is tremendous fun. Like music, it is a skill that derives from an unknown blend of innate talent and constant practice. Like drawing, it can be shaped to a variety of ends – commercial, artistic, and pure entertainment. Programmers have a well-deserved reputation for working long hours, but are rarely credited with being driven by creative fevers. Programmers talk about software development on weekends, vacations, and over meals not because they lack imagination, but because their imagination reveals worlds that others cannot see.

Larry O'Brien and Bruce Eckel in *Thinking in C#*

Graphs - definition

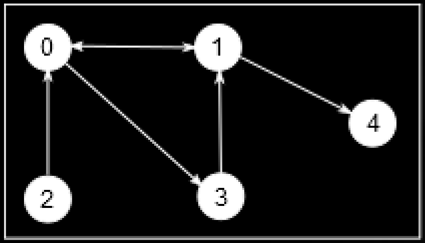
- A **graph** is a representation of a set of objects where some pairs of objects are connected by links
  - The objects are called vertices, or nodes
  - The links are called edges
- The edges of a graph can be directed or undirected, resulting in a directed or undirected graph


Undirected Graph


Directed Graph

Graphs – data structures

- Most common way to store graph information is through an adjacency matrix

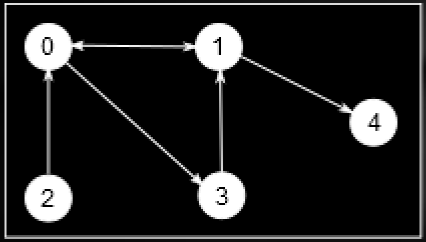


$$\begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} = \mathrm{M}$$

- $M[i, j] == 1$if and only if there is an edge between vertex i and j
- Adjacency matrices for undirected graphs are symmetrical
- If the graph has a large amount of vertices, but few edges, then the adjacency matrix becomes sparse (most elements are 0) thus other ways are used to store the information

Graphs – data structures (cont'd)

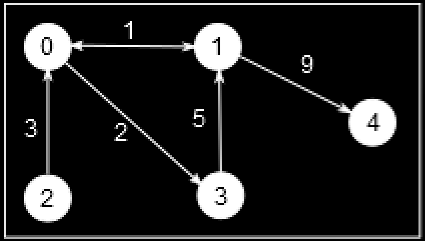
- Another way to store a graph with few edges is through a list of edges, where an edge is defined as a pair of nodes



((0,1), (0,3), (1,0), (1,4), (2,0), (3,1))

Graphs – data structures (cont'd)

- In most real-life applications for graphs, edges are used to model routes between locations (cities, stations, data-centers, etc.); in this case, some routes are better than others so graph edges may have an associated cost

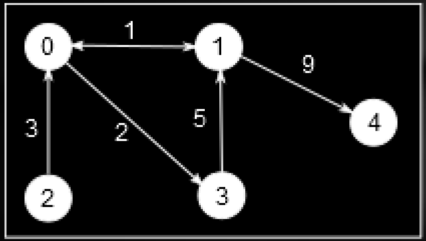


$$\begin{pmatrix} -1 & 1 & -1 & 2 & -1 \\ 1 & -1 & -1 & -1 & 9 \\ 3 & -1 & -1 & -1 & -1 \\ -1 & 5 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \end{pmatrix} = \mathrm{M}$$

- In this case, the adjacency matrix will hold the edge cost, or -1 if there is no link between the vertices; in some applications, it is possible that instead of -1 to have a very large value in order to model a very high cost for trying to travel on that route

Graphs – data structures (cont'd)

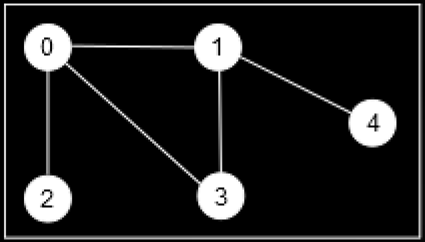
- In the case of lists, the edges are now defined by the nodes, and the cost



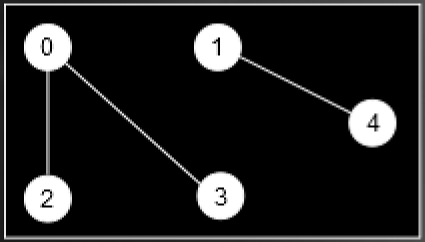
((0,1,1), (0,3,2), (1,0,1), (1,4,9), (2,0,3), (3,1,5))

Graphs - algorithms

- Two vertices in a graph are called connected if there is a path between them.
- An undirected graph is connected any two vertices are connected.
- A directed graph is called strongly connected, if any two vertices in the graph are connected
- A directed graph is called weakly connected if by replacing all directed edges with undirected edges, the graph becomes connected.



Connected graph



Unconnected graph

Exercise: Given an undirected graph defined through its adjacency matrix, devise an algorithm to ascertain if the graph is connected.

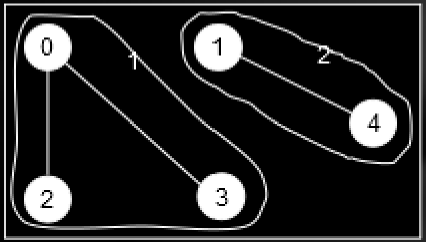
Graphs - algorithms

- Breadth-first and Depth-first search can be applied to graphs as well as trees
- Breadth-first search is implemented using a queue
- Depth-first search is implemented using a stack

Exercise: Write a program that parses a graph defined by its adjacency matrix, depth first, then breadth first.

Graphs - algorithms

- A connected component of a graph is defined as a sub-graph of that graph, which is connected.
- A sub-graph of a graph is a graph defined by a subset of the original graph's vertices and associated edges.



A graph with two connected components

Exercise: Adapt the previous algorithm in order to count the connected components of an undirected graph.

Thank you!